WHAT IS CLAIMED IS:

1. A high performance data engine comprising:

(a) a computer system including a central processing unit, a memory and programming logic; and

(b) a data table comprising a plurality of data structures, wherein each of said data structures includes at least a first portion and a second portion, said first portion storing a data value, said second portion storing a length value associated with the data value,

wherein, when the plurality of data structures are sorted in the data table, if a first data value comprising N digits and a second data value comprising at least N+1 digits comprise an identical string of digits for the first N digits, then the first data value precedes the second data value in the sorted data table,

wherein if a third data value comprising M digits and a fourth data value comprising at least M digits comprise an identical string of digits only for the first Q digits where $Q < M$, then the ordering of the third and fourth data values is determined by the ordering of the $(Q+1)^{st}$ digit from the third data value and the $(Q+1)^{st}$ digit from the second data value.

2. The high performance data engine of claim 1, wherein the first portion precedes the second portion in the data structure.

3. The high performance data engine of claim 1, wherein the data table further comprises a plurality of rows, each of said rows comprising a set of data structures.

4. The high performance data engine of claim 3, wherein each of said rows further comprises a start range value, wherein said start range value identifies a first member of the set of data structures comprising each of said rows.

5. The high performance data engine of claim 3, wherein each of said rows further comprises an end range value, wherein said end range value identifies a last member of the set of data structures comprising each of said rows.

6. The high performance data engine of claim 3, wherein each of said rows further comprises a start range value and an end range value, wherein said start range value identifies a first member of the set of data structures comprising each of said rows and , wherein said end range value identifies a last member of the set of data structures comprising each of said rows.

7. A high performance data engine comprising:

(a) a computer system including a central processing unit, a memory and programming logic; and

(b) a data table comprising a plurality of data structures, wherein each of said data structures includes at least a first portion and a second portion, said first portion storing a data value, said second portion storing a length value associated with the data value,

wherein, when the plurality of data structures are sorted in the data table, if a first data value comprises N digits and a second data value comprises at least N+1 digits, then the first data value precedes the second data value in the sorted data table,

wherein, if a third data value and a fourth data value both comprise M digits, and if the third data value and the fourth data value comprise an identical string of digits only for the first Q digits where $Q < M$, then the ordering of the first and second data values is determined by the ordering of the $(Q+1)^{st}$ digit from the first data value and the $(Q+1)^{st}$ digit from the second data value.

8. The high performance data engine of claim 7, wherein the second portion precedes the first portion in the data structure.

9. The high-performance data engine of claim 1 for use in data processing operations in the field of telephony.

10. The high-performance data engine of claim 1 wherein the data strings comprise telephone numbers.

11. The high-performance data engine of claim 1 wherein the data strings comprise postal routing codes.

12. A high performance data engine comprising:

(a) a computer system including a central processing unit, a memory and programming logic; and

(b) a plurality of data structures, wherein each of said data structures includes at least a first portion and a second portion, said first portion storing a data string, said second portion storing a length value associated with the data string;

wherein, each data structure has a predetermined fixed width, and

wherein the data string in each data structure represents a digit string of length zero or more, up to a predetermined maximum length, wherein said maximum length corresponds to a number of sub-fields provided in the first portion of the data structure.

13. The high performance data engine of claim 12, wherein each of said data structures comprises an unsigned integer used in manipulation and comparison operations in a computing process.

14. The high-performance data engine of claim 12 for use in data processing operations in the field of telephony.

15. The high-performance data engine of claim 12 wherein the data strings comprise telephone numbers.

16. The high-performance data engine of claim 12 wherein the data strings comprise postal routing codes.

17. The high performance data engine of claim 12, wherein each of said data structures comprises a multi-word unsigned integer used in manipulation and comparison operations in a computing process.

18. The high performance data engine of claim 12, wherein the length value follows the digit string the data structure so that it is considered less significant in a comparison operation, yielding a sorting order for a first digit string having a first length value and a second digit sting having a second length value wherein:

(a) if the first digit string is identical to second digit string, for the first n digits of the first and the second digit strings, where n is the minimum of the first and the second length values, then the digit string having shorter length value precedes the data string having the longer length value; otherwise

(b) if the first and second digit strings differ anywhere in the first n digits of the first and the second data strings, then the sorting order of the first and second strings is determined by an ordering of a leftmost non-matching correspondingly positioned digit in the first and second data strings.

19. The high performance data engine of claim 12, wherein the length value precedes the data value so that it is considered more significant in a comparison operation, yielding a sorting order for a first digit string having a first length value and a second digit sting having a second length value wherein:

(a) if the first length value is not equal to the second length value, then the digit string having shorter length string precedes the digit string having the longer length, otherwise

(b) if the first length value is equal to the second length value and if the first digit string is not equal to the second digit string, then the ordering of the first and second digit strings is determined by an ordering of a leftmost non-matching, correspondingly positioned, digit in the first and second digit strings

20. The high performance data engine of claim 12, wherein the plurality of data structures are organized into a plurality of rows in a data table, wherein each row comprises a set of

data structures and a start range value identifying a first member of the set of data structures comprising each of said rows.

21. The high-performance data engine of claim 20, wherein each start range value is stored in a data structure in a row.

22. The high-performance data engine of claim 21 wherein the length value follows the digit string the data structure so that it is considered less significant in a comparison operation, yielding a sorting order for a first digit string having a first length value and a second digit sting having a second length value wherein:

(a) if the first digit string is identical to second digit string, for the first n digits of the first and the second digit strings, where n is the minimum of the first and the second length values, then the digit string having shorter length value precedes the data string having the longer length value; otherwise

(b) if the first and second digit strings differ anywhere in the first n digits of the first and the second data strings, then the sorting order of the first and second strings is determined by an ordering of a leftmost non-matching correspondingly positioned digit in the first and second data strings.

23. The high-performance data engine of claim 20, wherein an ambiguous digit string is detected using a special comparison operation with a candidate matching row or a row following the candidate matching row.

24. The high-performance data engine of claim 20, further comprising an end range value identifying a last member of the set of data structures comprising each of said rows.

25. The high-performance data engine of claim 24, wherein each end range value stored in a data structure in a row.

26. The high-performance data engine of claim 25 wherein the length value follows the digit string the data structure so that it is considered less significant in a comparison

operation, yielding a sorting order for a first digit string having a first length value and a second digit sting having a second length value wherein:

(a) if the first digit string is identical to second digit string, for the first n digits of the first and the second digit strings, where n is the minimum of the first and the second length values, then the digit string having shorter length value precedes the data string having the longer length value; otherwise

(b) if the first and second digit strings differ anywhere in the first n digits of the first and the second data strings, then the sorting order of the first and second strings is determined by an ordering of a leftmost non-matching correspondingly positioned digit in the first and second data strings.

27. The high-performance data engine of claim 24, wherein an ambiguous digit string is detected using a special comparison operation with a candidate matching row or a row following the candidate matching row.

28. The high performance data engine of claim 12, wherein the plurality of data structures are organized into a plurality of rows in a data table, wherein each row comprises a set of data structures and an end range value identifying a last member of the set of data structures comprising each of said rows.

29. The high-performance data engine of claim 28, wherein each end range value is stored in a data structure in a row.

30. The high-performance data engine of claim 29, wherein the length value follows the digit string the data structure so that it is considered less significant in a comparison operation, yielding a sorting order for a first digit string having a first length value and a second digit sting having a second length value wherein:

(a) if the first digit string is identical to second digit string, for the first n digits of the first and the second digit strings, where n is the minimum of the first and the second length

values, then the digit string having shorter length value precedes the data string having the longer length value; otherwise

(b) if the first and second digit strings differ anywhere in the first n digits of the first and the second data strings, then the sorting order of the first and second strings is determined by an ordering of a leftmost non-matching correspondingly positioned digit in the first and second data strings.

31. The high-performance data engine of claim 28, wherein an ambiguous digit string is detected using a special comparison operation with a candidate matching row or a row following the candidate matching row.

32. The high performance data engine of claim 12, wherein the plurality of data structures are organized into a plurality of rows in a data table, wherein each row comprises a set of data structures and a start range value identifying a first member of the set of data structures comprising each of said rows and an end range value identifying a last member of the set of data structures comprising each of said rows.

33. The high-performance data engine of claim 32, wherein each start range value and each end range value is stored in a data structure in a row.

34. The high-performance data engine of claim 33, wherein the length value follows the digit string the data structure so that it is considered less significant in a comparison operation, yielding a sorting order for a first digit string having a first length value and a second digit sting having a second length value wherein:

(a) if the first digit string is identical to second digit string, for the first n digits of the first and the second digit strings, where n is the minimum of the first and the second length values, then the digit string having shorter length value precedes the data string having the longer length value; otherwise

(b) if the first and second digit strings differ anywhere in the first n digits of the first and the second data strings, then the sorting order of the first and second strings is determined by an ordering of a leftmost non-matching correspondingly positioned digit in the first and second data strings.

35. The high-performance data engine of claim 32, wherein an ambiguous digit string is detected using a special comparison operation with a candidate matching row or a row following the candidate matching row.

36. The high-performance data engine of claim 32 for use in data processing operations in the field of telephony.

37. The high-performance data engine of claim 32, wherein the data strings comprise telephone numbers.

38. The high-performance data engine of claim 32, wherein the data strings comprise postal routing codes.

39. A method for high performance data processing, said method comprising:

storing a plurality of data structures in a data table, wherein each of said data structures includes at least a first portion and a second portion, said first portion storing a data value, said second portion storing a length value associated with the data value; and

sorting the data table wherein, if a first data value comprising N digits and a second data value comprising at least N+1 digits comprise an identical string of digits for the first N. digits, then the first data value precedes the second data value in the sorted data table,

wherein, if a third data value comprising M digits and a fourth data value comprising at least M digits comprise an identical string of digits only for the first Q digits where Q < M, then the ordering of the first and second data values is determined by the ordering of the $(Q+1)^{st}$ digit from the first data value and the $(Q+1)^{st}$ digit from the second data value.

40. The method of claim 39, wherein the first portion precedes the second portion in the data structure.

41. The method of claim 39, wherein the data table further comprises a plurality of rows, each of said rows comprising a set of data structures.

42. The method of claim 41, wherein each of said rows further comprises a start range value, wherein said start range value identifies a first member of the set of data structures comprising each of said rows.

43. The method of claim 41, wherein each of said rows further comprises an end range value, wherein said end range value identifies a last member of the set of data structures comprising each of said rows.

44. The method of claim 41, wherein each of said rows further comprises a start range value and an end range value, wherein said start range value identifies a first member of the set of data structures comprising each of said rows and , wherein said end range value identifies a last member of the set of data structures comprising each of said rows.

45. A method for high performance data processing, said method comprising:

creating a data table comprising a plurality of data structures, wherein each of said data structures includes at least a first portion and a second portion, said first portion storing a data value, said second portion storing a length value associated with the data value; and

sorting the data table wherein, if a first data value comprises N digits and a second data value comprises at least N+1 digits, then the first data value precedes the second data value in the sorted data table, and

wherein, if a third data value and a fourth data value both comprise M digits, and if the third data value and the fourth data value comprise an identical string of digits only for the first Q digits where $Q < M$, then the ordering of the third and fourth data values is determined

by the ordering of the $(Q+1)^{st}$ digit from the third data value and the $(Q+1)^{st}$ digit from the fourth data value.

46. The method of claim 45, wherein the second portion precedes the first portion in the data structure.

47. The method of claim 45, wherein the data strings comprise telephone numbers.

48. The method of claim 45, wherein the data strings comprise postal routing codes.

49. A method for high performance data processing, said method comprising creating a plurality of data structures, wherein each of said data structures includes at least a first portion and a second portion, said first portion storing a data string, said second portion storing a length value associated with the data string wherein, each data structure has a predetermined fixed width, and wherein the data string in each data structure represents a digit string of length zero or more, up to a predetermined maximum length, wherein said maximum length corresponds to a number of sub-fields provided in the first portion of the data structure.

50. The method of claim 49, wherein each of said data structures comprises an unsigned integer used in manipulation and comparison operations in a computing process.

51. The method of claim 49, wherein the data strings comprise telephone numbers.

52. The method of claim 49, wherein the data strings comprise postal routing codes.

53. The method of claim 49, wherein each of said data structures comprises a multi-word unsigned integer used in manipulation and comparison operations in a computing process.

54. The method of claim 49, wherein the length value follows the digit string the data structure so that it is considered less significant in a comparison operation, yielding a sorting order for a first digit string having a first length value and a second digit sting having a second length value wherein:

(a) if the first digit string is identical to second digit string, for the first n digits of the first and the second digit strings, where n is the minimum of the first and the second length values, then the digit string having shorter length value precedes the data string having the longer length value, otherwise

(b) if the first and second digit strings differ anywhere in the first n digits of the first and the second data strings, then the sorting order of the first and second strings is determined by an ordering of a leftmost non-matching correspondingly positioned digit in the first and second data strings.

55. The method of claim 49, wherein the length value precedes the data value so that it is considered more significant in a comparison operation, yielding a sorting order for a first digit string having a first length value and a second digit sting having a second length value wherein:

(a) if the first length value is not equal to the second length value, then the digit string having shorter length string precedes the digit string having the longer length, otherwise

(b) if the first length value is equal to the second length value and if the first digit string is not equal to the second digit string, then the ordering of the first and second digit strings is determined by an ordering of a leftmost non-matching, correspondingly positioned, digit in the first and second digit strings.

56. The method of claim 49, wherein the plurality of data structures are organized into a plurality of rows in a data table, wherein each row comprises a set of data structures and a start range value identifying a first member of the set of data structures comprising each of said rows.

57. The method of claim 56, wherein each start range value is stored in a data structure in a row.

58. The method of claim 57, wherein the length value follows the digit string the data structure so that it is considered less significant in a comparison operation, yielding a sorting order for a first digit string having a first length value and a second digit sting having a second length value wherein:

(a) if the first digit string is identical to second digit string, for the first n digits of the first and the second digit strings, where n is the minimum of the first and the second length values, then the digit string having shorter length value precedes the data string having the longer length value; otherwise

(b) if the first and second digit strings differ anywhere in the first n digits of the first and the second data strings, then the sorting order of the first and second strings is determined by an ordering of a leftmost non-matching correspondingly positioned digit in the first and second data strings.

59. The method of claim 49, wherein an ambiguous digit string is detected using a special comparison operation with a candidate matching row or a row following the candidate matching row.

60. The method of claim 49, further comprising an end range value identifying a last member of the set of data structures comprising each of said rows.

61. The method of claim 60, wherein each end range value stored in a data structure in a row.

62. The method of claim 61 wherein the length value follows the digit string the data structure so that it is considered less significant in a comparison operation, yielding a sorting order for a first digit string having a first length value and a second digit sting having a second length value wherein:

(a) if the first digit string is identical to second digit string, for the first n digits of the first and the second digit strings, where n is the minimum of the first and the second length

values, then the digit string having shorter length value precedes the data string having the longer length value; otherwise

(b) if the first and second digit strings differ anywhere in the first n digits of the first and the second data strings, then the sorting order of the first and second strings is determined by an ordering of a leftmost non-matching correspondingly positioned digit in the first and second data strings.

63. The method of claim 60, wherein an ambiguous digit string is detected using a special comparison operation with a candidate matching row or a row following the candidate matching row.

64. The method of claim 49, wherein the plurality of data structures are organized into a plurality of rows in a data table, wherein each row comprises a set of data structures and an end range value identifying a last member of the set of data structures comprising each of said rows.

65. The method of claim 64, wherein each end range value is stored in a data structure in a row.

66. The method of claim 65, wherein the length value follows the digit string the data structure so that it is considered less significant in a comparison operation, yielding a sorting order for a first digit string having a first length value and a second digit sting having a second length value wherein:

(a) if the first digit string is identical to second digit string, for the first n digits of the first and the second digit strings, where n is the minimum of the first and the second length values, then the digit string having shorter length value precedes the data string having the longer length value, otherwise

(b) if the first and second digit strings differ anywhere in the first n digits of the first and the second data strings, then the sorting order of the first and second strings is determined

by an ordering of a leftmost non-matching correspondingly positioned digit in the first and second data strings.

67. The high-performance data engine of claim 28, wherein an ambiguous digit string is detected using a special comparison operation with a candidate matching row or a row following the candidate matching row.

68. The method of claim 49, wherein the plurality of data structures are organized into a plurality of rows in a data table, wherein each row comprises a set of data structures and a start range value identifying a first member of the set of data structures comprising each of said rows and an end range value identifying a last member of the set of data structures comprising each of said rows.

69. The method of claim 68, wherein each start range value and each end range value is stored in a data structure in a row.

70. The method of claim 69, wherein the length value follows the digit string the data structure so that it is considered less significant in a comparison operation, yielding a sorting order for a first digit string having a first length value and a second digit sting having a second length value wherein:

(a) if the first digit string is identical to second digit string, for the first n digits of the first and the second digit strings, where n is the minimum of the first and the second length values, then the digit string having shorter length value precedes the data string having the longer length value; otherwise

(b) if the first and second digit strings differ anywhere in the first n digits of the first and the second data strings, then the sorting order of the first and second strings is determined by an ordering of a leftmost non-matching correspondingly positioned digit in the first and second data strings.

71. The method of claim 68, wherein an ambiguous digit string is detected using a special comparison operation with a candidate matching row or a row following the candidate matching row.

72. The method of claim 68, wherein the data strings comprise telephone numbers.

73. The method of claim 68, wherein the data strings comprise postal routing codes.